

## 5 Lab: LMS Algorithm (15 pts)

In this lab question, you will implement the LMS algorithm with NumPy to learn a linear regression model for the provided dataset. You will also be directed to analyze how the choice of learning rate in the LMS algorithm affect the final result. [All the codes generating the results of this lab should be gathered in one file and submit to Gradescope.](#)

### Lab 1 (15 pts)

To start with, please download the `dataset.mat` file from Canvas and load it into NumPy arrays<sup>a</sup>. There are two variables in the file: data  $X \in \mathbb{R}^{100 \times 3}$  and target  $D \in \mathbb{R}^{100 \times 1}$ . Each individual pair of data and target is composed into  $X$  and  $D$  following the same way as discussed in Lecture 2. Specifically, each row in  $X$  corresponds to the transpose of a data point, with the first element as constant 1 and the other two as the two input features  $x_{1k}$  and  $x_{2k}$ . The goal of the learning task is finding the weight vector  $W \in \mathbb{R}^{3 \times 1}$  for the linear model that can minimize the MSE loss, which is also formulated on Lecture 2.

- (a) (3pt) Directly compute the least square (Wiener) solution with the provided dataset. What is the optimal weight  $W^*$ ? What is the MSE loss of the whole dataset when the weight is set to  $W^*$ ?
- (b) (4pt) Now consider that you can only train with 1 pair of data points and target each time. In such a case, the LMS algorithm should be used to find the optimal weight. Please initialize the weight vector as  $W^0 = [0.53, 0.20, 0.10]^T$ , and update the weight with the LMS algorithm. After each *epoch* (every time you go through all the training data and loop back to the beginning), compute and record the MSE loss of the current weight on the whole dataset. Run LMS for 20 epochs with learning rate  $r = 0.005$ , report the weight you get in the end, and plot the MSE loss in log scale vs. Epochs.
- (c) (3pt) Scatter plot the points  $(x_{1k}, x_{2k}, d_k)$  for all 100 data-target pairs in a 3D figure<sup>b</sup>, and plot the lines corresponding to the linear models you got in (a) and (b) respectively in the same figure. Observe if the linear models fit the data well.
- (d) (5pt) Learning rate  $r$  is an important hyperparameter for the LMS algorithm, as well as for CNN optimization. Here, try repeat the process in (b) with  $r$  set to 0.01, 0.05, 0.1 and 0.5 respectively. Together with the result you got in (b), plot the MSE losses of all sets of experiments in log scale vs. Epochs in one figure. Then try further enlarge the learning rate to  $r = 1$  and observe how the MSE changes. Based on these observations, comments on how the learning rate affects the speed and quality of the learning process. (Note: The learning rate tuning for the CNN optimization will be introduced in Lecture 7.)

<sup>a</sup>You may refer to <https://docs.scipy.org/doc/scipy/reference/generated/scipy.io.loadmat.html> for loading matrices in .mat file into NumPy arrays.

<sup>b</sup>Please refer to <https://jakevdp.github.io/PythonDataScienceHandbook/04.12-three-dimensional-plotting.html> for plotting 3D plots with Matplotlib.